Bayesian neural networks with confidence estimations applied to data mining

R. Orre, A. Lansner

SANS, Dept. of Computer Science, Royal Institute of Technology,
S-100 44 Stockholm, Sweden

A. Bate, M. Lindquist

WHO Collaborating Centre for International Drug Monitoring,
Uppsala Monitoring Centre, S-753 20 Uppsala,Sweden

Abstract

An international database of case reports, each one describing a possible case of adverse drug reactions (ADRs), is maintained by the Uppsala Monitoring Centre (UMC), for the WHO international program on drug safety monitoring. Each report can be seen as a row in a data matrix and consists of a number of variables, like drugs used, ADRs, and other patient data. The problem is to examine the database and find significant dependencies which might be signals of potentially important ADRs , to be investigated by clinical experts. We propose a method by which estimated frequencies of combinations of variables are compared with the frequencies that would be predicted assuming there were no dependencies. The estimates of significance are obtained with a Bayesian approach via the variance of posterior probability distributions. The posterior is obtained by fusing a prior distribution (Dirichlet of dimension 2^{n-1}) with a batch of data, which is also the prior used when the next batch of data arrives. To decide whether the joint probabilities of events are different from what would follow from the independence assumption, the "*information component*" $\log(P_{ij}/(P_iP_j))$ plays a crucial role, and one main technical contribution reported here is an efficient method to estimate this measure, as well as the variance of its posterior distribution, for large data matrices. The method we present is fundamentally an artificial neural network denoted Bayesian Confidence Propagation Neural Network (BCPNN). We also demonstrate an efficient way of finding complex dependencies. The method is now (autumn 1998) being routinely used to produce warning signals on new unexpected ADR associations .

1 Introduction

The fundamental aim is to find new unexpected dependencies between variables in a database. The database, which this methodology has been implemented on, consists of case reports of adverse drug reactions, reported from 50 WHO collaborating national centres. This database currently contains nearly two million reports, in each report more than 77 variable fields may be considered for analysis.

The database is updated quarterly with approximately 35000 reports. Primarily we want to find new unexpected associations in the data set occurring due to this quarterly update of the database. Initially between drugs or combinations of drugs and adverse reactions or combinations of adverse reactions, but also including other variables like country and patient age.

For this purpose we have extended a Bayesian neural network [LE89], [HL95] to be able to do estimations of variances of weights and posterior distributions to be suitable for data mining. The Bayesian neural network we use here is a feed forward network [1] where the learning and inference rules are based upon Bayes rule [Bay63] ,[Lap14] for conditional probabilities. We want to find the posterior probability function for an outcome or *response variable* A which is conditioned by a joint input event or *explanatory variable* D under the assumption that we can express the joint likelihood density $P(D|A)$ as a product of n independent marginal densities $P(d_i|A)$ as

$$P(A|D) = P(A)\frac{P(D|A)}{P(D)} = P(A)\frac{P(d_1|A) \cdot P(d_i|A) \cdots P(d_n|A)}{P(D)}. \qquad (1)$$

The outcome A may in general be represented by a continuous distribution, but here we deal with discrete outcomes. In the following A means the set of mutually exclusive outcomes $a_1, a_j, \ldots, a_m$. We use the symbol "A" as this often represents *adverse drug reactions* or a combinations thereof in our application. The input events d_i most often represents drugs or combinations of drugs. Such a network is fundamentally a naive Bayesian classifier [Goo50] but it has earlier been extended with higher order units that deals with classification and diagnosis also for tasks involving dependent inputs [LH96], where it was denoted

[1] it has also been used as a recurrent Hopfield-like network useful for pattern completion [LE85],[LE89],[Kon89]

BCPNN (Bayesian Confidence Propagation Neural Network). Here we extend the latter by calculating also the variance

$$V\left(\frac{P(a_j|d_i)}{P(a_j)}\right) = V\left(\frac{P(d_i, a_j)}{P(d_i)P(a_j)}\right) \tag{2}$$

which is a particularly useful measure, for instance, when we do data mining, particularly on associations with low frequency counts, where the uncertainty may be large. Similarly we calculate $V(P(a_j|D))$, the variance for a posterior probability, which gives us a confidence measure of a prediction or classification task. Following established practice in the area of Bayesian neural networks, we use $E(P(A|D))$, $V(P(A|D))$, etc to denote the mean and variance of the posterior distributions of $P(A|D)$.

The Bayesian feed forward neural networks have similarities to Bayesian Belief Networks [Pea88] and they can theoretically be transformed into each other [HL95]. The main difference is that in the latter only the dependent variables are dealt with in each node, whereas in the neural network model both dependent and independent variables are treated in parallel and the result is propagated through a few layers only.

1.1 Bayesian Inference in BCPNN

Let the response variable A be composed of m *mutual exclusive* outcome events a_j. Bayes rule gives the following relation

$$P(a_j|D) = \frac{P(a_j)P(D|a_j)}{P(D)} = \frac{P(a_j)P(D|a_j)}{\sum_j P(a_j)P(D|a_j)}. \tag{3}$$

Then, let the joint explanatory event D be composed of n *independent* events d_i, such that

$$P(D|a_j) = P(d_1|a_j) \cdot P(d_2|a_j) \cdots P(d_n|a_j).$$

Each event d_i is further made up of K_i mutually exclusive sub-events, or states, such that

$$P(d_i|a_j) = P(d_i^1|a_j) + P(d_i^2|a_j) \cdots P(d_i^{K_i}|a_j).$$

These assumptions, i.e. $P(d_i^k|a_j)$ being independent over i and mutual exclusive over k gives

$$P(D|a_j) = \prod_i \sum_k P(d_i^k|a_j), \tag{4}$$

which, using Bayes rule, can be rewritten as

$$P(D|a_j) = \prod_i \sum_k \frac{P(a_j|d_i^k)}{P(a_j)} P(d_i^k). \tag{5}$$

Now, replace $P(a_j|d_i^k)$ above with $\frac{P(d_i^k,a_j)}{P(d_i^k)} = \frac{P(a_j,d_i^k)}{P(d_i^k)}$ (definition) and $P(d_i^k)$ with its *belief* value π_i^k, which expresses the current belief on event d_i^k during training and inference in an *exhaustive* way, *i.e.* $[\pi_{d_i^k} \geq 0, \sum_{k=1}^{k=K_i} \pi_i^k = 1]$. A binary variable is thus represented by one "on-unit" and one "off-unit". We get

$$P(D|a_j) = \prod_i \sum_k \frac{P(d_i^k, a_j)}{P(d_i^k)P(a_j)} \pi_i^k, \tag{6}$$

which we generally consider approximately valid also when $P(d_i^k|a_j)$ are *almost independent* over i. Using equation (6), equation (3) can now be written

$$P(a_j|D) \propto P(a_j) \prod_i \sum_k \frac{P(d_i^k, a_j)}{P(d_i^k)P(a_j)} \pi_i^k, \tag{7}$$

which resembles many *feed forward* artificial neural network architectures. For discrete belief values ($\pi_{d_i^k} \in \{0,1\}$) we may use the following simplified form

$$P(a_j|D) \propto \exp\left[\log P(a_j) + \sum_i \sum_k \log\left[\frac{P(d_i^k, a_j)}{P(d_i^k)P(a_j)}\right] \pi_{d_i^k}\right]. \tag{8}$$

In the last expression (8) we would recognize "exp" as the *transfer function* and "$\log P(A)$" as a *bias* term from other artificial neural network architectures. The corresponding weight value is then either $[\log \frac{P(d_i^k,A)}{P(d_i^k)P(A)}]$ as in (8) or just $[\frac{P(d_i^k,A)}{P(d_i^k)P(A)}]$ as in (7). Which equation to prefer depends on the application. For precise mixture modelling of *e.g.* continuous variables [OL96] preferably (7) be used, due to the better accuracy, rather than (8). The logarithmic form in (8) has been used a lot in *e.g.* recurrent networks [LE89] and also in classification [HL95]. In the data mining application described here and in [BLE$^+$98] we use the logarithmic form, as this has a nice connection with information theory, especially mutual information [Pea88]. Therefore we refer to the term $[\log \frac{P(d_i^k,A)}{P(d_i^k)P(A)}]$ as *information component* because it is a measure of the information that migrates from one state of a variable to one state of another variable. Mutual information in its discrete form can then be regarded as a weighted sum of *information components*,

$$I(X;Y) = \sum_x \sum_y P(x,y) \log \frac{P(x,y)}{P(x)P(y)}. \tag{9}$$

In the rest of this paper we use the following definitions of the weights and information components (observe that index k is assumed but not always included throughout this text):

$$W_{ij} = \frac{P(d_i^k, a_j)}{P(d_i^k)P(a_j)}, \tag{10}$$

$$IC_{ij} = \log W_{ij}. \tag{11}$$

2 Method to estimate probabilities and uncertainties

We start by estimating the probability for a single binary event of a Bernoulli trial represented by a variable with outcomes 0 and 1. The likelihood function for c_1, *i.e.* the probability to get c_1 number of outcome 1 from a total of $C = c_0 + c_1$ trials, is a binomial distribution:

$$P(c_1|p_1, C) = \binom{C}{c_1} p_1^{c_1}(1-p_1)^{c_0}. \tag{12}$$

In the classical perspective we get the maximum of the likelihood by differentiating vs p_1 and solve $\frac{d}{dp_1}P(c_1|p_1) = 0$ as

$$\begin{aligned} c_1(1-p_1) &= c_0 p_1 \\ \hat{p_1} &= \frac{c_1}{c_0+c_1} = \frac{c_1}{C}. \end{aligned} \tag{13}$$

This classical estimate does, however, not give us accurate estimates of p_1 for small counter values and does not tell us anything about the significance of an estimated probability. To overcome this we use the Bayesian method to assert an *a priori* probability distribution for the variable, which is refined when more *information i.e.* samples, become available. We consider p_1 to be drawn from a conjugate family of distributions, which we assert as the prior distribution. A convenient prior which is much used if we do not expect the input to be a multi modal mixture [BS94],[Hec97] is the Beta distribution, described by hyperparameters α_1 and α_0

$$P(p_1) = \frac{\Gamma(\alpha_1 + \alpha_0)}{\Gamma(\alpha_1)\Gamma(\alpha_0)} p_1^{\alpha_1 - 1}(1 - p_1)^{\alpha_0 - 1} \tag{14}$$

which gives a posterior for p_1, given the counters c_1 and c_0, which is also a Beta [BS94]:

$$P(p_1|c_1, c_0) = \frac{\Gamma(C + \alpha_1 + \alpha_0)}{\Gamma(c_1 + \alpha_1)\Gamma(c_0 + \alpha_0)} p_1^{c_1 + \alpha_1 - 1}(1 - p_1)^{c_0 + \alpha_0 - 1}. \tag{15}$$

The expectation value $\hat{p_1} = E(p_1)$ we get by integration and normalization, where the reduction makes the Γs disappear:

$$E(p_1) = \frac{\int_0^1 p_1 \cdot {p_1}^{c_1 + \alpha_1 - 1}(1 - p_1)^{c_0 + \alpha_0 - 1} dp}{\int_0^1 {p_1}^{c_1 + \alpha_1 - 1}(1 - p_1)^{c_0 + \alpha_0 - 1} dp}. \tag{16}$$

The solution to this [where $Beta(x, y) = \Gamma(x)\Gamma(y)/\Gamma(x + y)$] is:

$$E(p_1) = \frac{Beta(c_1 + 1 + \alpha_1, C - c_1 + \alpha_0)}{Beta(c_1 + \alpha_1, C - c_1 + \alpha_0)} \tag{17}$$

which simplified gives the following ($\alpha = \alpha_1 + \alpha_0$) for $\hat{p_1}$,

$$\hat{p_1} = E(p_1) = \frac{c_1 + \alpha_1}{C + \alpha}. \tag{18}$$

In the same way we find the variance estimation ($\hat{\sigma_p^2} = V(p) = E(p^2) - E(p)^2$) and the estimate of $V(p_1)$ becomes

$$V(p_1) = \frac{(c_1 + \alpha_1)\,(C - c_1 + \alpha - \alpha_1)}{(C + \alpha)^2\,(1 + C + \alpha)}. \tag{19}$$

2.1 Joint Probabilities

As a prior for the joint probability p_{ij}, which has four different outcomes, we assert a 3-dimensional Dirichlet-distribution of p_{11}, p_{10} and p_{01} ($p_{00} = 1 - p_{11} - p_{10} - p_{01}$) in the hyperparameters $\gamma_{11}, \gamma_{10}, \gamma_{01}, \gamma_{00}$. Consider *e.g.* the distribution of P_{11} [$P(p_{11})$]:

$$\begin{aligned} P(p_{11}) &= Di(p_{11}|\gamma_{11}, \gamma_{10}, \gamma_{01}, \gamma_{00}) \qquad (20)\\ &= \frac{\Gamma(\gamma_{11} + \gamma_{10} + \gamma_{01} + \gamma_{00})}{\Gamma(\gamma_{11})\Gamma(\gamma_{10})\Gamma(\gamma_{01})\Gamma(\gamma_{00})} {p_{11}}^{\gamma_{11}-1} {p_{10}}^{\gamma_{10}-1} {p_{01}}^{\gamma_{01}-1} (1 - p_{11} - p_{10} - p_{01})^{\gamma_{00}-1}. \end{aligned}$$

The marginal distributions to Dirichlet are also Dirichlet but in this case they reduce to a one dimensional Dirichlet which is a Beta (14). The posterior distribution given the counters $c_{11}, c_{10}, c_{01}, c_{00}$ is also a Dirichlet distribution [BS94]:

$$P(p_{11}|c_{11}, c_{10}, c_{01}, c_{00}) = Di(p_{11}|c_{11} + \gamma_{11}, c_{10} + \gamma_{10}, c_{01} + \gamma_{01}, c_{00} + \gamma_{00}).$$

The expectation value $E(p_{11})$ thus becomes:

$$E(p_{11}) = \frac{\int_0^1 \int_0^1 \int_0^1 p_{11} Di(p_{11}|c_{11} + \gamma_{11}, c_{10} + \gamma_{10}, c_{01} + \gamma_{01}, c_{00} + \gamma_{00}) dp_{01} dp_{10} dp_{11}}{\int_0^1 \int_0^1 \int_0^1 Di(p_{11}|c_{11} + \gamma_{11}, c_{10} + \gamma_{10}, c_{01} + \gamma_{01}, c_{00} + \gamma_{00}) dp_{01} dp_{10} dp_{11}}.$$

The evaluation of this integral involves some hyper-geometric functions and is a bit cumbersome and we skip the details here. These expectation values can also be looked up in a statistical textbook like [BS94]. We end up with the following:

$$E(p_{11}) = \frac{c_{11} + \gamma_{11}}{c_{11} + \gamma_{11} + c_{10} + \gamma_{10} + c_{01} + \gamma_{01} + c_{00} + \gamma_{00}} = \frac{c_{11} + \gamma_{11}}{C + \gamma} \tag{21}$$

and for the variance (observe similarity with (18,19))

$$V(p_{11}) = \frac{E(p_{11})(1 - E(p_{11}))}{1 + c_{11} + \gamma_{11} + c_{10} + \gamma_{10} + c_{01} + \gamma_{01} + c_{00} + \gamma_{00}} \\ = \frac{(c_{11} + \gamma_{11})(C + \gamma - c_{11} - \gamma_{11})}{(C+\gamma)^2(1 + C + \gamma)}. \tag{22}$$

2.2 Weights and Information Components

In our first attempt to find the expectation values for the weights $[E(W_{ij}) = E(\frac{p_{ij}}{p_i p_j})]$ and their variances we tried the same approach as above by using the integral:

$$\int_0^1 \int_0^1 \int_0^1 \frac{{p_{11}}^{\gamma_{11}-1} {p_{10}}^{\gamma_{10}-1} {p_{01}}^{\gamma_{01}-1} (1 - p_{11} - p_{10} - p_{01})^{\gamma_{00}-1}}{(p_{11} + p_{10})^{\gamma_{11}+\gamma_{10}-2} (p_{11} + p_{01})^{\gamma_{11}+\gamma_{01}-2}} dp_{11} dp_{10} dp_{01}. \tag{23}$$

We could, however, not find any closed form solution to this, which would still not have taken into account any cross dependencies. Instead the following approximation is used, utilizing (18),(21) above, where α and β are the number of mutually exclusive events in each class for the variables i and j respectively

$$E(W_{ij}) \approx \frac{E(p_{ij})}{E(p_i)E(p_j)} = \frac{(c_{ij} + \gamma_{ij})(C + \alpha)(C + \beta)}{(C + \gamma)(c_i + \alpha_i)(c_j + \beta_j)}. \tag{24}$$

For the specific case of the IC_{ij} this can, however be calculated exactly, due to

$$E(IC_{ij}) = E(\log \frac{p_{ij}}{p_i p_j}) = E(\log p_{ij}) - E(\log p_i) - E(\log p_j) \tag{25}$$

and it can be shown [KO98] that when p is Beta(a, b) distributed, then

$$E(\log p) = \frac{b}{a(a+b)} - b \cdot \sum_{n=1}^{\infty} \frac{1}{(a+n)\cdot(a+b+n)}. \tag{26}$$

Here $a = \alpha_1 + c_1$ and $b = \alpha_0 + c_0$. In the application work we present here we have, however, used the following simplified form for the expectation value $E(IC_{ij})$

$$E(IC_{ij}) \approx \log E(W_{ij}) \approx \log \frac{E(p_{ij})}{E(p_i)E(p_j)}. \tag{27}$$

The variance for the weight $[V(W_{ij}) = E(W_{ij}^2) - E(W_{ij})^2]$ is harder to estimate. So far we have used the Gauss' approximation for the variance of a function *i.e.* $V[g(X_1, \ldots, X_k)] \approx \sum_{i=k}^{k} V(X_i)(\frac{\partial g}{\partial \mu_i})^2$, and not included covariant terms.

We assume symmetrical distributions, therefore we set $\mu_i = E(X_i)$. The variance for the weight $V(W_{ij})$ then is

$$\begin{aligned} V(W_{ij}) \approx & \frac{V(p_{ij})}{\hat{p}_i^2 \hat{p}_j^2} + \frac{\hat{p}_{ij}^2 V(p_i)}{\hat{p}_i^4 \hat{p}_j^2} + \frac{\hat{p}_{ij}^2 V(p_j)}{\hat{p}_i^2 \hat{p}_j^4} & (28) \\ = & \frac{(C+\alpha)^2 (C+\beta)^2 (c_{ij}+\gamma_{ij})}{(C+\gamma)^2 (c_i+\alpha_i)^2 (c_j+\beta_j)^2} & (29) \\ & \cdot \left[\frac{(C - c_{ij} + \gamma - \gamma_{ij})}{(1 + C + \gamma)} + \frac{(c_{ij}+\gamma_{ij})(C - c_i + \alpha - \alpha_i)}{(c_i + \alpha_i)(C + \alpha + 1)} + \frac{(c_{ij}+\gamma_{ij})(C - c_j + \beta - \beta_j)}{(c_j + \beta_i)(C + \beta + 1)} \right]. & \end{aligned}$$

For the information component IC_{ij} we can, due to the properties of the log-function as in (25) and in [KO98] write the variance $V(IC_{ij})$ as an exact expression (here including covariant terms):

$$\begin{aligned} V(IC_{ij}) = & V(\log p_{ij}) + V(\log p_{ij}) + V(\log p_{ij}) & (30) \\ & -2cov(\log p_{ij}, \log p_i) - 2cov(\log p_{ij}, \log p_j) + 2cov(\log p_i, \log p_j) & \end{aligned}$$

and it can be proved [KO98] that for p being Beta(a, b) distributed, then

$$V(\log p) = \sum_{n=0}^{\infty} \frac{b^2 + 2ab + 2bn}{(a+n) \cdot (a+b+n)^2}. \tag{31}$$

For $V(IC_{ij})$ we intend using expression(30) with covariant terms in the future. Although we are still using the simpler approach with Gaussian approximation, assuming independence:

$$V(IC_{ij}) \approx V(p_{ij}) \left(\frac{1}{\hat{p}_{ij}}\right)^2 + V(p_i) \left(\frac{-1}{\hat{p}_i}\right)^2 + V(p_j) \left(\frac{-1}{\hat{p}_j}\right)^2. \tag{32}$$

Here we measure the IC_{ij} in bits (i.e. use $\log_2$), which gives the following explicit expression

$$V(IC_{ij}) \approx \frac{\dfrac{C - c_{ij} + \gamma - \gamma_{ij}}{(c_{ij} + \gamma_{ij})\,(1 + C + \gamma)} + \dfrac{C - c_i + \alpha - \alpha_i}{(c_i + \alpha_i)(1 + C + \alpha)} + \dfrac{C - c_j + \beta - \beta_i}{(c_j + \beta_j)(1 + C + \beta)}}{(\log 2)^2}. \quad (33)$$

2.3 Variance of Conditioned Posterior Distribution

To calculate the variance of $P(a_j|D)$, below, we do a logarithmic exponential transformation, using Gaussian approximation for variance of a function $[V(g(X)) \approx V(X) \cdot (\frac{\delta g}{\delta X}(E(X)))^2]$ thus $V(X) = V(e^{\log[X]}) \approx V(\log[X]) \cdot E(X)^2$. An approximate variance for a sum of independent terms, (38), can then be calculated using the Gaussian approximation $V(\sum_i c_i \cdot X_i) \approx \sum_i c_i^2 \cdot V(X_i)$.

Let $\theta(a_j|D)$ below, be the expression after the independence assumption (4). When $\theta(a_j|D)$ is scaled by the coefficient $1/\kappa = 1/\sum_j \theta(a_j|D)$, we obtain the expression for the posterior $P(a_j|D)$, i.e. $\sum_j P(a_j|D) = 1$. From equations (3) and (6) we get

$$P(a_j|D) = \frac{P(a_j)P(D|a_j)}{\sum_j P(a_j)P(D|a_j)} = \frac{\theta(a_j|D)}{\sum_j \theta(a_j|D)} = \frac{\theta(a_j|D)}{\kappa} \quad (34)$$

$$\theta_j = \theta(a_j|D) = P(a_j) \prod_i \sum_k \frac{P(d_i^k, a_j)}{P(d_i^k)P(a_j)} \pi_{d_i^k} \quad (35)$$

$$\kappa = \sum_j \theta(a_j|D). \quad (36)$$

By using a Taylor expansion the variance $V(P(a_j|D))$ can be approximated as

$$V(P(a_j|D)) \approx \frac{V(\theta_j)}{E(\kappa)^2} - 2 \cdot \frac{cov(\theta_j, \kappa)E(\theta_j)}{E(\kappa)^3} + \frac{V(\kappa)E(\theta_j)^2}{E(\kappa)^4}. \quad (37)$$

The variance $V(\kappa)$ is zero, due to κ being a function of the applied data pattern only (3). For a similar reason the covariance $cov(\theta_j, \kappa)$ is also zero, as we only consider d_i^k to be a random variable during the training phase of the network.

In (38), below, we start with $V(\log[\theta(a_j|D)])$. We set $[W_{ij}^k = \frac{P(d_i^k,a_j)}{P(d_i^k)P(a_j)}]$ in (39) and consider $\log[P(a_j)]$ and $\sum_k W_{ij}^k \cdot \pi_{d_i^k}$ to be independent in (40). A logarithmic variance transformation is performed from (40) to (41). The $\pi_{d_i^k}$ represents part of a mixture of k belief values, which are here coefficients only, without a variance. As $\pi_{d_i^k}$ is part of a mixture $[\sum_k \pi_{d_i^k} = 1]$ it is reasonable to assume that $2 \cdot \sum_{k<l} \pi_i^k \pi_i^l cov(W_{ij}^k, W_{ij}^l) \leq 0$. The Gaussian approximation for a sum of independent variables will then be a worst case estimate of (41), which results in the inequality (42).

$$\begin{aligned}
V(\log[\theta(a_j|D)]) &= V\left(\log[P(a_j)] + \sum_i \log\left[\sum_k \frac{P(d_i^k,a_j)}{P(d_i^k)P(a_j)}\pi_{d_i^k}\right]\right) && (38)\\
&= V\left(\log[P(a_j)] + \sum_i \log\left[\sum_k W_{ij}^k \cdot \pi_{d_i^k}\right]\right) && (39)\\
&\approx V\left(\log[P(a_j)]\right) + \sum_i V\left(\log\left[\sum_k W_{ij}^k \cdot \pi_{d_i^k}\right]\right) && (40)\\
&\approx V\left(\log[P(a_j)]\right) + \sum_i \frac{V\left(\sum_k W_{ij}^k \cdot \pi_{d_i^k}\right)}{E\left(\sum_k W_{ij}^k \cdot \pi_{d_i^k}\right)^2} && (41)\\
&< V\left(\log[P(a_j)]\right) + \sum_i \frac{\sum_k V\left(W_{ij}^k\right) \cdot \pi_{d_i^k}^2}{E\left(\sum_k W_{ij}^k \cdot \pi_{d_i^k}\right)^2}. && (42)
\end{aligned}$$

When we know that $\pi_{d_i^k}$ represents k *mutually exclusive* discrete inputs, then we can rewrite (40) as (43), because $\sum_k$ is then a sum over one single value only, as all other values are zero. Therefore we can move $\sum_k$ and $\pi_{d_i^k}$ outside the variance expression, which is done in (44). The $\left[\log W_{ij}^k\right]$ we have earlier named the *information component* $\left[IC_{ij}^k\right]$ (10) and for that we have an exact variance expression (30) and for $V(log[P(a_j)])$ as well (31), which gives equation (45).

$$\begin{aligned}
V(\log[\theta(a_j|D)]) &\approx V\left(\log[P(a_j)]\right) + \sum_i V\left(\sum_k \log\left[W_{ij}^k\right] \cdot \pi_{d_i^k}\right) && (43)\\
&\approx V\left(\log[P(a_j)]\right) + \sum_i \sum_k V\left(\log\left[W_{ij}^k\right]\right) \cdot \pi_{d_i^k} && (44)\\
&= V\left(\log[P(a_j)]\right) + \sum_i \sum_k V\left(IC_{ij}^k\right) \cdot \pi_{d_i^k}. && (45)
\end{aligned}$$

Using either (42) or (45), depending on the type of input events, we can calculate $V(\log(\theta(a_j|D)))$. The variance $V(\theta(a_j|D))$ thus becomes

$$V(\theta(a_j|D)) = V(\exp(\log[\theta(a_j|D)])) \approx V(\log \theta(a_j|D)) \cdot E(\theta(a_j|D))^2. \tag{46}$$

The expression (37) for the variance $V(P(a_j|D))$ can thus be written

$$V(P(a_j|D)) \approx V(\log[\theta(a_j|D)]) \cdot \frac{E(\theta(a_j|D))^2}{E(\sum_j \theta(a_j|D))^2}. \tag{47}$$

2.4 The Selection of Reasonable Priors

The priors for p_i and p_j are not critical as the convergence to the "real probability" is rather quick [BS94]. The most simple prior for a binary variable to use here is $\alpha_1 = \alpha_0 = 1$, *i.e.* a non-informative (sometimes called *ignorant prior*) [BS94], which corresponds to an a priori assumption about equal probability distribution. We should, however, be aware that for a non binary discrete variable with k states, we need another prior where each specific state of the variable is considered. We could then assert for example $\alpha_i = 1, \alpha = k \cdot \alpha_i$; $\beta_j = 1, \beta = l \cdot \beta_j$. To get a coherent prior for p_{ij} we would then set $\gamma_{ij} = 1, \gamma = \alpha \cdot \beta$. We have, however, chosen to make a slight drawback from the coherence criterion and instead chosen a prior for p_{ij} that behaves well for small counter values in the way that when we have no data samples of pairs of variables we consider them to be independent (48) and then choose the prior for p_{ij} so that (49) is fulfilled.

$$\lim_{c_i, c_j, c_{ij} \to 0} IC_{ij} = \log \frac{\hat{p}_{ij}}{\hat{p}_i \hat{p}_j} \approx 0 \tag{48}$$

$$\lim_{c_{ij}, C \to 0} IC_{ij} = \log \frac{\hat{p}_{ij}}{\hat{p}_i \hat{p}_j} = \log \frac{\frac{c_{ij}+\gamma_{ij}}{C+\gamma}}{\hat{p}_i \hat{p}_j} \approx 0. \tag{49}$$

Then we can set $\gamma_{ij} = 1$ and $\gamma = \frac{\gamma_{ij}}{\hat{p}_i \hat{p}_j}$

In figure 1 we see an example of how this looks when starting with the ignorant prior and then how the posterior distribution gets more and more narrow when we add a few samples. In the figure it is also shown how the estimated prior for the joint distribution P_{ij} look like for some of the first samples.

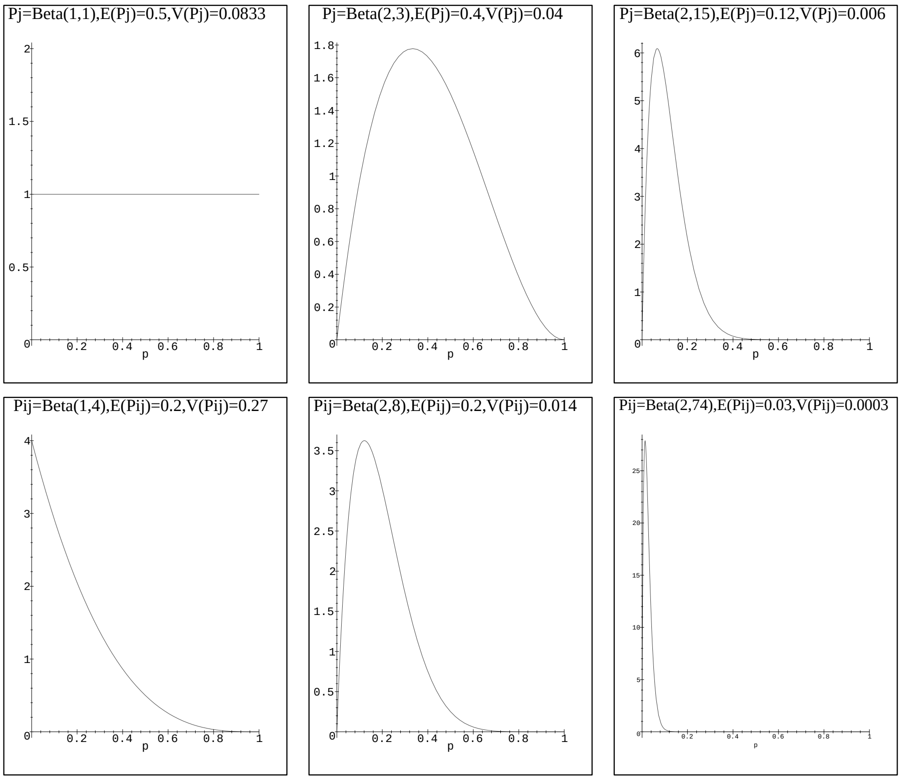

Fig. 1. Some examples of prior distributions for $P(p_i)$ and $P(p_{ij})$ (briefly P_i and P_{ij} in the diagrams above). The **upper diagrams** show the priors for $P(p_i)$ when $[\alpha_0 = \alpha_1 = 1]$ $(c_0 = 0, c_1 = 0)$; $(c_0 = 2, c_1 = 1)$ and $(c_0 = 14, c_1 = 1)$ respectively. The **lower diagrams** show the corresponding estimated priors for $P(p_{11})$ when $[\gamma_{11} = 1]$ $c_{11} = 0, c_{11} = 1, c_{11} = 1$ when both i, j are defined as above.

3 OTHER METHOD RELATED ISSUES

3.1 Variable Value Coding

The coding of binary and discrete variables into a *neural layer* representation is rather straight forward. For a binary variable x we input the values $[x, \bar{x}]$. This is also the simplest example of a *hypercolumn*, a neural layer with mutually exclusive input units. When a variable value is missing we may input the a priori probabilities $[p_x, p_{\bar{x}}]$, alternatively we code it as a default value. Any discrete variable, whose values are mutually exclusive is coded in this way. In general we may input a normalized mixture of belief values instead of the a priori probabilities for missing values. Real valued variables are coded using such a mixture of belief values, which represents the degree of membership to a set of Radial Basis Functions (RBFs). The placement of these RBFs is usually done using the EM algorithm [Trå93].

3.2 Dependent Variables

Variables which are found to be dependent on each other can be handled by coding the combination of states for these variables into a separate subspace, a hypercolumn, a neural layer where all combinations are mutually exclusive. As an example, assuming that the binary variables X and Y are dependent on each other, we make a hypercolumn with 4 units representing $\{\bar{x}\bar{y}, \bar{x}y, x\bar{y}, xy\}$. When the number of combinations get large this coding may be inconvenient, then a *reduced* coding is used, where only the combinations or *features* that actually occur in training data are coded. Real valued variables are handled in a similar way. The RBF units, which are then used, may combine an arbitrary number of dimensions into, which can be seen as, a normalized mixture of belief values as one hypercolumn. The RBF coding is done "before" the treatment of discrete variables, which is necessary to be able to combine real valued variables with discrete variables. To find dependencies between variables we use the following methods (ϕ is a threshold parameter):

- Check for strong pairwise mutual information between all pairs of subspaces. The procedure is repeated to find higher order combinations as long as

$$\phi_{MI} < \sum_{xy} p_{xy} \log \frac{p_{xy}}{p_x p_y}. \tag{50}$$

- Check all variable combinations up to a certain complexity level in one shot. To decide what combinations to save we use the Kullback-Leibler distance between the joint distribution and the marginal distributions. We save those subspaces where

$$\phi_{KL3\ldots} < \sum_{xyz\ldots} p_{xyz\ldots} \log \frac{p_{xyz\ldots}}{p_x p_y p_z \cdots}. \tag{51}$$

- Check all combinations of variables up to a certain complexity level in one run. Save those combinations only, where the *information component* between input and output layer is above a certain threshold θ_{IC} which is

$$\phi_{IC} < \log \frac{p_{xy}}{p_x p_y}. \tag{52}$$

A comment on the thresholds ϕ_{MI} , ϕ_{KL} and ϕ_{IC} above: At the moment these kind of thresholds are considered to be design parameters in the BCPNN network. We have no automatic method for generating the threshold levels yet.

3.3 Sparse Matrix Technique

When working with this huge WHO database, of adverse drug reactions, we first used full matrixes. We found this to be inefficient because a typical full connection matrix, containing 20-50 million connection elements often contained a non zero value in 1-2% of the positions only. Therefore we developed a sparse matrix technique, which reduced both the required computer time as well as memory requirements drastically as it does not create matrix elements until they are needed. The technique is “double sparse” *i.e.* it allows us to create not only matrix elements dynamically, but also the “neurons” in the input/output layers dynamically.

Thanks to this sparse technique and the organization of the database in reports, where only a very small subset of all possible combinations can occur on each, we do not need to do the search for dependent variables completely incrementally. We can decide beforehand how high a complexity level of combinations we want to investigate.

4 RESULTS AND EXAMPLES

4.1 Signal Generation

In the database application we want to generate an early warning signal when a certain dependency between a drug or a set of drugs and an adverse drug reaction (ADR) or a set of ADRs is detected. The procedure is to look for significant differences in weight values between input and output variables when a batch of reports is added to the database. To be able to do this in an efficient manner from the perspective of computing time and memory 0 we used a sparse matrix coding of the connection matrix. The procedure was tested on some well known signals like the association between the drug suprofen and back pain, and azapropazone-photosensitivity. Results from these time scans are shown in figure 2...5 respectively.

In these experiments the BCPNN network was set up in the normal way *i.e.* C is the total number of reports in the database, c_i is the number of reports for the drug, c_j is the number of reports for the adverse reaction and c_{ij} is the number of reports where the drug and the adverse reaction occur on the same report.

The diagram in figure 2 shows how the IC *(information component)* for the suprofen-back pain association varies between the years 1983 and 1990. The bars around the IC curve show, for each quarterly year on the x-axis, a 95 % confidence interval for the IC. The diagram in figure 3 shows how the cumulative probability function for IC being greater than zero $[P(IC > 0) = \int_{y=0}^{\infty} P_y(IC)dy]$ develops over time. A case report of acute flank pain after taking 3 doses of suprofen was first published in 1986 [HM86].

From the diagrams in figure 2 and figure 3 we can see indications of an association between the drug and the adverse reaction with rather high certainty, around 80 % after the first quarter 1984, which rises to around 97% in the middle of 1984, when we would signal it. The current criterion for the detection of a signal is when the lower 95 % confidence limit of the IC for the drug-ADR combinations changes from a negative to a positive value on addition of the data for the last quarter.

For the azapropazone case there was a paper published in 1985 of this drug being associated with photosensitivity reaction [OBB85]. The diagram in figure 4, which shows the IC for azapropazone vs photosensitivity reaction, indicates this association would be highlighted with this approach in 1975. In figure 5 we see the prior probabilities for the drug and the adverse reaction. Observe that the scale for the probabilities in this diagram is logarithmic. We also see the posterior probability for the adverse reaction given the drug. As can be seen $P(A|D) >> P(A)$, which clearly indicates a conditioned dependency between the drug and the ADR. All three probabilities are shown with 95% confidence intervals, but the prior probabilities are much narrower than the conditioned posterior probability because there are less samples in the joint distribution.

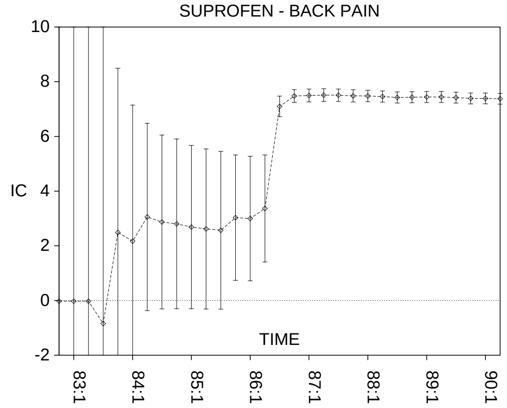

Fig. 2. A well known signal: suprofen and back pain. The diagram shows the IC (*information component*) for the drug-ADR association. The error bars show a 95% estimated conf. interval.

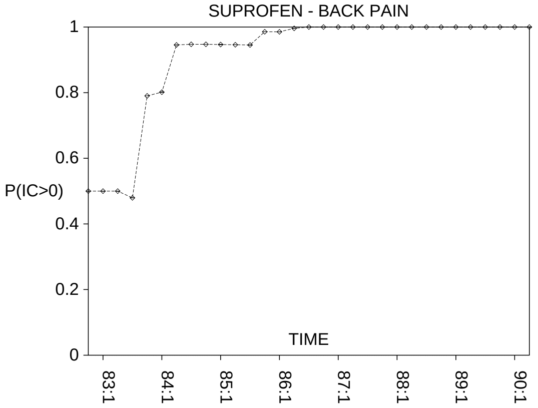

Fig. 3. Suprofen and back pain: The diagram shows how the $P(IC > 0)$ develops over time, we see a clear indication of this association with 80% certainty already after the first quarter 1984.

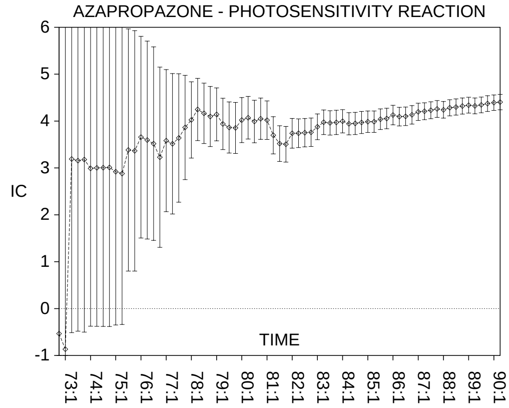

Fig. 4. The development from 1973 to 1990 of the information component for the drug azapropazone vs the photosensitivity reaction with 95% conf. int.

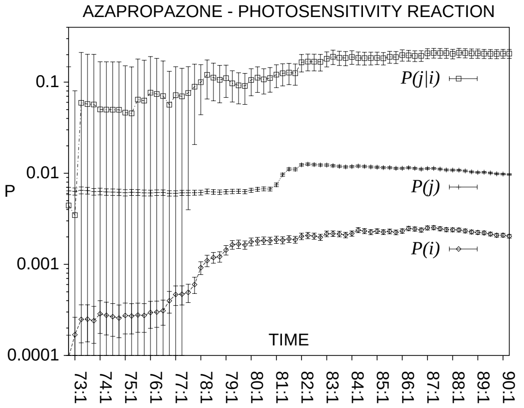

Fig. 5. Logarithmic scale, with 95% conf. int. Priors: $P(i) = P(azapropazone)$, $P(j) = P(photosensitivity)$. Posterior: $P(j|i) = P(photosensitivity|azapropazone)$

4.2 Digoxin versus Age and Rash

The following experiments aim to demonstrate that IC analysis can be used to study the relationship between combinations of any variables in the database, including, but not being restricted to, drug adverse reaction association pairs. To establish this, the relationship between the drug digoxin and the patient's age was examined by observing the change in the IC for the association digoxin-rash over different age intervals. Also the strength of association between digoxin and age intervals was examined for the entire database.

The BCPNN network was here set up to generate counters in a slightly different manner than previously. C would normally be the total number of reports in the database, but in two of these experiments (in figure 8 and 9) it was set as C=total number of reports within the specific age group under consideration. The age grouping used here are 10 year intervals. The c_i is the counter for the drug or for the drug combined with age. The occurrences of the drug being reported as "suspected" (figure 8) or concomitant medication ("other") (figure 9) are counted separately. c_j= the number of reports for the adverse reaction or the adverse reaction combined with age group, and c_{ij}=counts the intersection between c_i and c_j in the normal way.

In figure 6 we see a normal time scan of the IC for digoxin versus the adverse reaction Rash from 1967 to 1997, when the IC has stabilized at a level of -2. The diagram in figure 8 shows the IC for the database up to the end of 1997, but here displayed separately for different age groups. In this diagram (figure 8) we see, for each age interval, that there was a negative IC between digoxin and rash. The association was most negative for age range $30 \cdots 40$, although in general there seemed to be a trend towards lower ICs for higher ages, *i.e.* less probability for digoxin to be the suspected drug for causing Rash in elderly patients. However, the confidence intervals are rather large and the trend is therefore unreliable, based on the data available at the time. We can also see that the uncertainty in IC is higher for younger patients, which may be explained by the diagram in figure 7, where we see how the IC for digoxin vs age varies with the age of the patient. The diagram of IC vs age in figure 7 shows a clear trend of increasing IC with age. From a minimum of IC=-4 for $20 \cdots 30$ year olds ($c_{ij} = 34$) to a maximum value of IC=3 ($c_{ij} = 244$) for the age group of 90+ year olds. The highest c_{ij} value was for $70 \cdots 80$ year old patients where ($c_{ij} = 2228$) (IC=1.7). The standard deviations are small for all IC values due to the large number of reports of digoxin in the database (7370).

In figure 9 the IC between digoxin and Rash within different age groups is shown when digoxin was not the suspected drug but was reported as concomitant medication. In the same way as for the results where digoxin was the suspected drug most digoxin-rash associations had negative ICs , however there was a definite trend of increasing positive IC for increasing age range, so that for age groups $70 \ldots 80$, $80 \ldots 90$ and 90+ there was a definite positive association between digoxin, when recorded as concomitant medication, and rash.

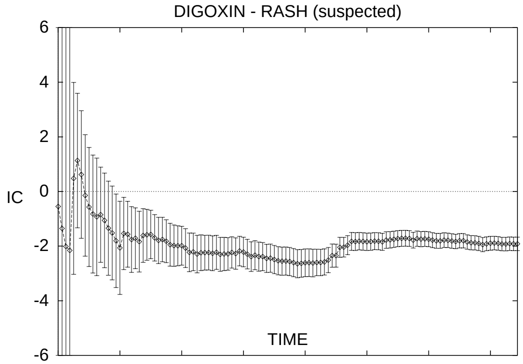

Fig. 6. A time scan of IC for the drug digoxin vs the ADR rash from the year 1967 to 1997. At 1997 the IC has stabilized around a level of −2.

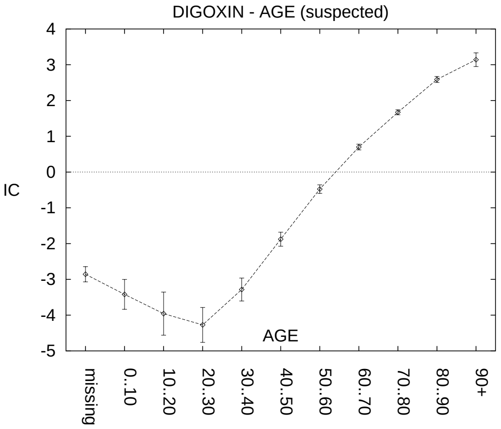

Fig. 7. Here we see how the association between digoxin and patient age varies with the age of the patient. Thus indicating a higher probability to find elderly patients being reported with digoxin than younger patients.

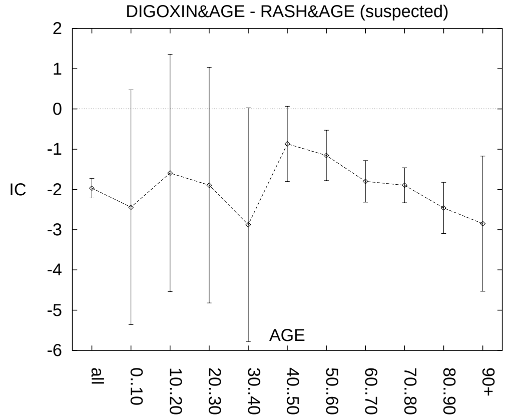

Fig. 8. IC between digoxin and rash for the last quarter 1997 displayed separately for each age group, with ten year intervals. The age group "all" sums all intervals.

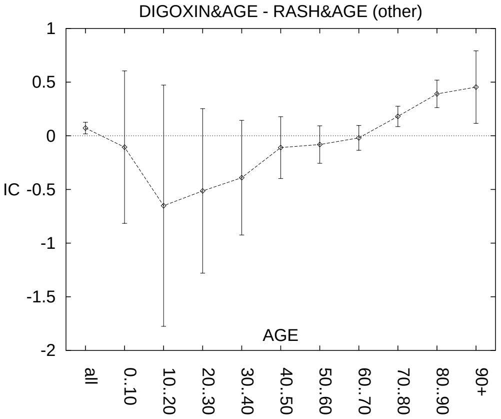

Fig. 9. The IC between digoxin and rash displayed separately for each age group, last quarter 1997. This concerns when digoxin was reported as concomitant medication ("other").

4.2.1 Discussion

The contrast between digoxin-rash profiles over age for digoxin as a suspected drug and digoxin as concomitant medication was striking. This is probably because when elderly patients are take digoxin they are more likely to be taking other drugs concomitantly, than a younger person taking digoxin. Therefore the occurrence of rash as an adverse reaction is more likely to be attributed to another drug, for elderly patients as compared to younger patients. This experiment shows that the BCPNN methodology can be used to look at associations when 3 different variables are considered together.

5 SEARCH FOR DEPENDENT VARIABLES

5.1 Goal Description

The BCPNN methodology can be used to search for dependent variable combinations of any order. Here we provide an example using this method to data mine the database to assess the validity of our complex variable method and to provide an indication of the effectiveness of the methodology in finding combined variable effects. In this example, a syndrome, which is a group of concurrent symptoms is investigated. This experiment also shows that the method is computationally tractable.

We considered a known adverse drug reaction syndrome complex association: Neuroleptic Malignant Syndrome (NMS) which is frequently reported in the WHO database, and is mainly associated with antipsychotic drugs. The syndrome itself is a combination of several symptoms which themselves can be reported as individual adverse drug reactions. The following adverse reactions were selected as indicators: Creatine Phosphokinase Increased, Fever, Death and Hypertonia. Although death is an outcome it was included because it is also a "reportable" term in the adverse reaction terminology.

We are interested therefore in the associations between combinations of these adverse reactions with haloperidol, an antipsychotic drug known to cause NMS, and how the strength of the associations with the combinations compare with the strength of associations with the adverse reactions themselves. Ultimately we wish to know whether it will be possible to pick the syndrome out, not knowing the constituent reactions beforehand.

We aimed to investigate all single, pair and triplet combinations of all adverse reactions in the database and then examine the strength of the association of each combination with the drug haloperidol. For this purpose we used the *sparse matrix* (see 3.3) method which was an appropriate tractable method in this case. One conventional method to use would be to first scan the database to check all ADRs against all ADRs, then make a selection of what ADR pairs to consider based on some threshold. After this the database could be rescanned checking these selected ADR pairs versus all ADRs again. Then a new selection could be made by thresholding and these triplets checked versus the drug.

This unsupervised approach works well in finding general feature detectors as it would, in most cases, find combinations where the Kullback-Leibler distance between the joint adverse reactions density and the marginal product density

$$\sum_{123} p(ADR_{123}) \log \frac{p(ADR_{123})}{p(ADR_1)p(ADR_2)p(ADR_3)} \tag{53}$$

would be quite large, but this would not necessarily give us all the reactions we really want to find, i.e. all combinations where

$$p(ADR_{123}|drug) >> p(ADR_{123}). \tag{54}$$

The specification given the sparse BCPNN was to partition the drugs into the classes "haloperidol", "other drug" as input layer and make all possible combinations of adverse reactions in the output layer, *i.e.* the output layer will represent a subset of the power set of all ADRs on each report. The subset we used here included combinations of up to three ADRs. This would also allow us the to check *e.g.* the KL-distance (53) for all ADRs found in the database at once, which gave us a tremendous speedup compared with the original matrix approach, which took several days on a Sun UltraSparc. The actual search needed only about 7 hour of computing time on the same UltraSparc. At the same time we are able to consider all possible combinations of triplets of ADRs in the database to find those that satisfy: $p(ADR_{123}|drug) >> p(ADR_{123})$.

We generated lists of the associations according to the following table:

drug	ADR-comb	# ADR comb	# $IC > 0$
haloperidol	single-ADR	1700	281
haloperidol	double-ADR	35000	4019
haloperidol	triple-ADR	550000	5388

where the column "# ADR comb" tells us how many combinations that were found in total. The column "# $IC > 0$" tells us how many of these had a positive IC. The ones with a positive IC were then sorted on the level of the IC, *i.e.* the strength of the association between the drug and the ADR-combination.

As was expected the term NMS was on the top of all these lists. In the pairs and triplets list NMS was also found to be strongly associated with some of the other symptoms which are included in the symptom picture of the selected ADR terms. We also found that the selected ADRs where high on all three lists. For the single ADR list all four were among the highest 200 IC values. For the list with ADR pairs combinations of these four ADRs were also found among the highest 200 IC values, and three of these were in the top ten. For the list with triple ADRs all combinations with these four ADRs were among the highest 400 and three combinations were in the top ten.

6 DISCUSSION

This paper presents a new efficient methodology for data mining of large databases in a computationally feasible way. The method not only provides a way to calculate conditional dependencies and predictive posterior probabilities within the data, but also estimates of the variances of the corresponding distributions, which makes this method accurate also for small sample set sizes of the investigated variables within the data set. Although the method has been demonstrated on a specific database it is suitable for other data mining applications.

We have made extensive use of the Gaussian approximation formula for the variances of functions here. These approximations are best suited for Gaussian distributions, although here we use Beta and Dirichlet as our model distributions. To allow for this, particularly when the conditional independence assumption between input variables is not fulfilled, we code the dependent variables into hypercolumns, that is, partition the input space into mutually exclusive regions. Further on, to make the calculations simpler, we do not yet consider covariance terms in the calculations.

Initially it was difficult to do exact calculations of the expectation value and the variance of the IC_{ij}. It was therefore encouraging to find, as described in section 2.2 and also in [KO98], that using the logarithmic form of the IC_{ij} we can express these solutions in an exact analytical way. This is being considered in ongoing work.

In the results presented in this paper we have propagated probabilities and calculated the variances of posterior output distributions conditioned on a set of inputs by approximating the W_{ij}. We expect that the use of the analytical expression for the IC_{ij} will help us to make a better approximation of W_{ij}.

We intend to investigate the possibility of finding an exact expression for the conditioned output probability distribution, or at least, its variance. Alternatively we could approximate these variances reasonably by the use of numerical integration. This would, however, result in a very large increase in the computational power requirement. This can certainly be done if necessary and we will consider this approach in the future.

Although the neural network technology we use is not only computationally but also architecturally efficient other methodological approaches may be similarly applied to these variance calculations. There are statistical methods being developed that may do better in approximating the variances like "saddle point approximations for statistical series" [Kol97], which may be computationally feasible, but we have not been considered these yet. Our goal is to be able to propagate complete distributions and for this purpose sampling techniques, like Gibbs sampling, are often used today. This kind of technique is, however, at the moment, too computationally demanding to be used in our data mining application.

We believe that our approach provides a mechanism for earlier and more efficient signalling of suspected adverse drug reactions. This application is dealt with in more detail in [BLE$^+$98]. The method is now (spring 1998) being used to produce warning signals on new unexpected drug adverse reaction associations when they become significant in the database.

7 ACKNOWLEDGEMENTS

We gratefully acknowledge the reporting of spontaneous adverse drug reaction case reports from National Centres to the WHO Collaborating Centre for International Drug Monitoring, which make up the database described in this paper. Rogelio Melhado Defretias was the person that initialized the contact between WHO Uppsala monitoring centre and Royal Institute of Technology. Thanks to Anders Holst who supplied a lot of patience with us discussing the issues herein. He also provided the original idea of how to do the variance calculations using Gauss formula. Thanks to Stefan Arnborg for valuable comments.

References

[Bay63] Thomas Bayes. An essay towards solving a problem in the doctrine of chances. *Biometrika (reprint 1958 of orig art in Philos. Trans. R. Soc. London 53, pp. 370-418)*, 45:296–315, 1763.

[Bis95] Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, 1995.

[BLE$^+$98] Andrew Bate, Marie Lindquist, I. Ralph Edwards, Sten Olsson, Roland Orre, Anders Lansner, and Rogelio Melhado De Freitas. A bayesian neural network method for adverse drug reaction signal generation. *European Journal of Clinical Pharmacology*, 54:315–321, 1998.

[BS94] José M. Bernado and Adrian F.M. Smith. *Bayesian Theory.* John Wiley & Sons, Chichester, 1994.

[Goo50] I. J. Good. *Probability and the Weighing of Evidence.* Charles Griffin, London, 1950.

[Hec97] David Heckerman. Bayesian networks for data mining. *Data Minining and Knowledge Discovery*, 1:79–119, 1997.

[HL95] Anders Holst and Anders Lansner. A higher order bayesian neural network for classification and diagnosis. In *Applied Decision Technologies: Computational Learning and Probabilistic Reasoning*, pages 251–260, 1995. London, England, April 3-5.

[HM86] N.E. Henann and J.R. Morales. Suprofen - induced acute renal failure. *Drug Intell Clin Pharm*, 20(11):860–862, 1986.

[Hol97] Anders Holst. *Using Bayesian Neural Networks for Classification Tasks.* PhD thesis, Royal Institute of Technology, Stockholm, Sweden, Dept. of Numerical Analysis and Computing Science, 1997.

[KO98] Timo Koski and Roland Orre. Statistics of the information component in bayesian neural networks. Tech. Rep. TRITA-NA-9806, Dept. of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden, 1998.

[Kol97] John E. Kolassa. *Series Approximation Methods in Statistics.* Springer-Verlag, New York, 1997.

[Kon89] Igor Kononenko. Bayesian neural networks. *Biol. Cybernetics*, 61:361–370, 1989.

[Lap14] Pierre Simon Laplace. *A Philosophical Essay on Probabilities.* Trans 1995 from the 5th ed 1825 (orig 1814) by Andrew Dale. Orig "Essai Philisophique sur les Probabilités", Dover Publications, New York, 1814.

[LE85] Anders Lansner and Örjan Ekeberg. Reliability and speed of recall in an associative network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):490–498, July 1985.

[LE89] Anders Lansner and Örjan Ekeberg. A one-layer feedback artificial neural network with a bayesian learning rule. *International Journal of Neural Systems*, 1(1):77–87, 1989.

[LH96] Anders Lansner and Anders Holst. A higher order Bayesian neural network with spiking units. *International Journal of Neural Systems*, 7(2):115–128, May 1996.

[OBB85] S. Olsson, C. Biriell, and G. Boman. Photosensitivity during treatment with azapropazone. *Br Med J*, page 939, 1985.

[OL96] Roland Orre and Anders Lansner. Pulp quality modelling using bayesian mixture density neural networks. *Journal of Systems Engineering*, 6:128–136, 1996.

[Pea88] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, 1988.

[Trå93] Hans G.C. Tråvén. *On Pattern Recognition Applications of Artificial Neural Networks.* PhD thesis, Royal Institute of Technology, Stockholm, Sweden, Dept. of Numerical Analysis and Computing Science, 1993.